EVOLUTIONARY SYNTHESIS OF ANALOG ELECTRONICS USING SIMULATED ANNEALING

Josef Slezák
Doctoral Degree Programme (5), FEEC BUT
E-mail: xsleza08@stud.feec.vutbr.cz

Supervised by: Jiří Petržela
E-mail: petrzelj@feec.vutbr.cz

Abstract: The paper is focused on evolutionary synthesis of passive and active analog electronics circuits using simulated annealing global optimization method. Based on desired response as input impedance or voltage transfer, the method is able to synthesis topology and sizing of an analog circuit. The synthesis capability of the proposed method will be demonstrated in several examples of analog circuit synthesis problems.

Keywords: evolutionary electronics, simulated annealing, chaos, HSpice, fractal function

1 INTRODUCTION

Over the last two decades there has been published many papers focused on evolutionary electronics synthesis. Varieties of different optimization methods have been used. John Koza demonstrated suitability of genetic programming [1] however this method requires huge computation effort. Genetic algorithms and their modifications were the most used methods [2], [3]. R. Zebulum has presented 3 types of GA with variable length representation called OLG, ILG, UDIP [3]. Interesting work employing circuit building blocks has been presented by A. Das and R. Vemuri [4]. Presented paper is focused on evolutionary electronics synthesis using global optimization method simulated annealing (SA) which is very low used in the field of evolutionary electronics. Although there can be found papers where SA is used, it is usually employed only as a complement of another optimization method [5]. Presented work describes synthesis of analog electronic circuits by means of SA algorithm only.

2 SYNTHESIS OF ANALOG ELECTRONICS USING SIMULATED ANNEALING

Global optimization method Simulated annealing was described by S. Kirkpatrick in 1983 [6]. Pseudocode of used SA is presented in Figure 1.

```
k = 0; e = $e_i$                       / init counter k and solution e
    while T > $T_f$                     / termination criteria
    T = temp(k,$T_0$);                  / set temperature T
    i = rand<1,n>;                    / choose variable of e
    $\delta$x = rand{-1,1}                  / x set randomly to -1 or 1
    $e_n$ = e; $e_n$(i) = $e_n$(i) + $\delta$x        / generate neighborhood
    $\Delta$c = c($e_n$) – c(e)               / calculate delta of cost
    if Pa(T, $\Delta$c) > rand then         / test of acceptance
        e = $e_n$                        / $e_n$ accepted as new state e
    end                               /
    k = k+1                           / next iteration
end
```

Figure 1: Pseudocode of used simulated annealing method.

As you can see in Figure 1 the first step of SA is initialization of counter k and solution e which is set randomly. Thereafter main iteration loop begins. In every iteration temperature T is calculated according to exponential cooling scheme which is given by $T = T_0.exp(-r.k)$ [6]. Coefficient r defines speed of cooling. In the next step neighborhood e_n of actual state e is generated. Variable of solution e is randomly chosen and its value is, based on the result of random generator, increased or decreased by 1. After evaluation of cost of neighborhood $c(e_n)$, difference of cost of neighborhood and cost of old state can be computed. According to (1) neighborhood e_n is accepted as new state e or not. While current temperature T decreases this process repeats until T reaches defined final temperature T_f.

$$Pa = \frac{1}{1+exp(\Delta c/T)} \tag{1}$$

Encoding string e is formed of n integer numbers. For passive circuits every component of the circuit is defined by sequence of 5 integer variables. Therefore maximal complexity of encoded circuit is $n/5$. Method of encoding analog circuit by string of integer numbers which was adopted from [3] is described in Figure 2.

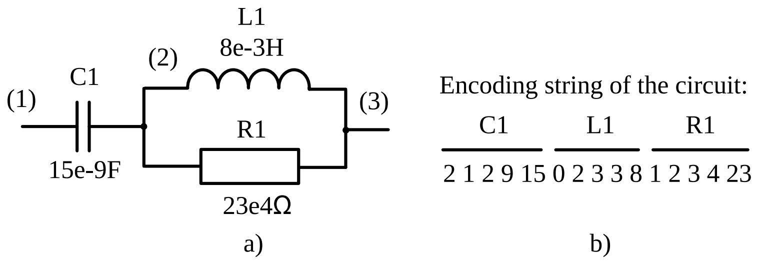


Figure 2: a) Simple analog circuit and b) its encoding string.

Every component of encoded circuit is described by sequence of 5 integer variables defining its type, connection node 1, connection node 2, exponent (*exp*) and mantisa (*man*). Definition of types of components is: 0 - L, 1 - R, 2 - C. Numbers (1), (2), (3) in the schematic denote names of the nodes. For example capacitor C1 which is connected between nodes 1 and 2 and has value 15nF ($C1 = man.10^{p.exp}$, where $p = -1$ for capacitor, $man = 15$ and $exp = 9$) is defined by sequence 2 1 2 9 15 (first five numbers in the encoding string in Figure 2b). This way other components of the circuit in Figure 2a can be encoded. Similar method allows encoding of all kinds of electrical components as for example transistors or operational amplifiers..

Result of the simulated circuit is calculated using nodal analysis in Matlab or using circuit simulator HSpice. In the case of HSpice, vector e has to be transformed into netlist representation first. The cost value is calculated according to (2) [3]

$$cost = \sum_{i=1}^{m} w_i(f_d(i) - f_a(i))^2, \tag{2}$$

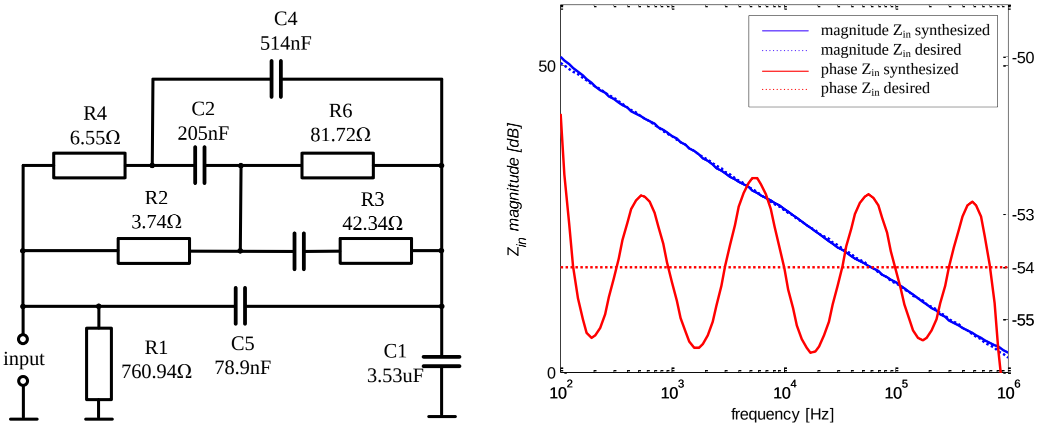
where m is number of points of response, w is weight vector, f_d is desired characteristic and f_a is response of actual solution. Approximate duration of one simulation is 5 ms for Matlab nodal analysis and 200 ms for HSpice. Although nodal analysis in Matlab is much faster it is suitable only for simulation of passive circuits (input impedance vs. frequency, voltage transfer vs. frequency). If real active components as for example BJTs or FETs are required in desired circuit, using of spice compatible circuit simulator is necessary. Nevertheless this results in longer run time of the optimization process.

In the next section there will be presented four different analog circuit synthesis problems.

3 EXAMPLES OF SOLVED PROBLEMS

3.1 SYNTHESIS OF NETWORK REALIZING FRACTION ORDER SYSTEM

The first presented problem is synthesis of analog circuit which realizes fraction order system [7] with input impedance $Z_{in} = 1,5849.10^4.s^{-0.6}$. The circuit was synthesized only using R, L, C components. Parameters of optimization: $T_0 = 6000, r = 7.10^{-5}, T_f = 3.10^{-10}, n = 50$ (maximally 10 components), $m = 101, evs = 440.10^3$ (cost function evaluations). Weight w was not used. The synthesis took 71 min on standard PC (Intel Core2 6420@2GHz, 1GB RAM). Cost was calculated in Matlab using nodal analysis. In Figure 3 there is schematic of synthesized circuit, its magnitude (blue lines) and phase (red lines) responses. Broken lines represent ideal curves of Z_{in}, solid lines represent responses of synthesized circuit. The maximal errors of magnitude and phase were $err_{mag} = 0.32$ dB at 3020 Hz and $err_{phas} = 1.69°$ at 5754 Hz respectively (errors in the boundaries of used frequency range were not considered). Compared to circuit obtained using classical circuit synthesis method in [7], proposed solution has better accuracy. Magnitude and phase errors of the circuit obtained in [7] were $err_{mag} = 0.62$ dB and $err_{phas} = 5.2°$.



(a) (b)

Figure 3: a) Schematic of synthesized network b) desired Z_{in} and Z_{in} of synthesized network

3.2 SYNTHESIS OF ADMITTANCE NETWORK FOR CHAOTIC OSCILLATOR

The second presented problem is synthesis of admittance network with input impedance (3). Parallel connection of this admittance network and PWL in Figure 4b realizes chaotic oscillator

$$Z_{in} = \frac{0.8410e^{-10}s^2 + 0.8160e^{-5}s - 0.3620e^{-1}}{0.1e^{-14}s^3 + 0.2890e^{-10}s^2 + 0.5980e^{-5}s - 0.92e^{-2}} \tag{3}$$

Admittance network was synthesized using R, L, C components only. Note that also negative resistors were allowed in this synthesis. Parameters of optimization: $T_0 = 2000, r = 3.10^4, T_f = 3.10^{-10}, n = 50, m = 61$. Weight vector was set $w = 1.4$ for range 100Hz to 184Hz , $w = 1.7$ for range 7.36kHz to 18.5kHz and $w = 1.4$ for range 117kHz to 1MHz , $evs = 100e3$. Run time was 11 min. Cost was calculated in Matlab using nodal analysis. Schematic of the synthesized admittance network is in Figure 4a. The magnitude and phase of (3) and of the synthesized network are presented in Figure 4c. As can be seen from the picture, the accuracy of the synthesis is very high. The state projection is shown in Figure 4d.

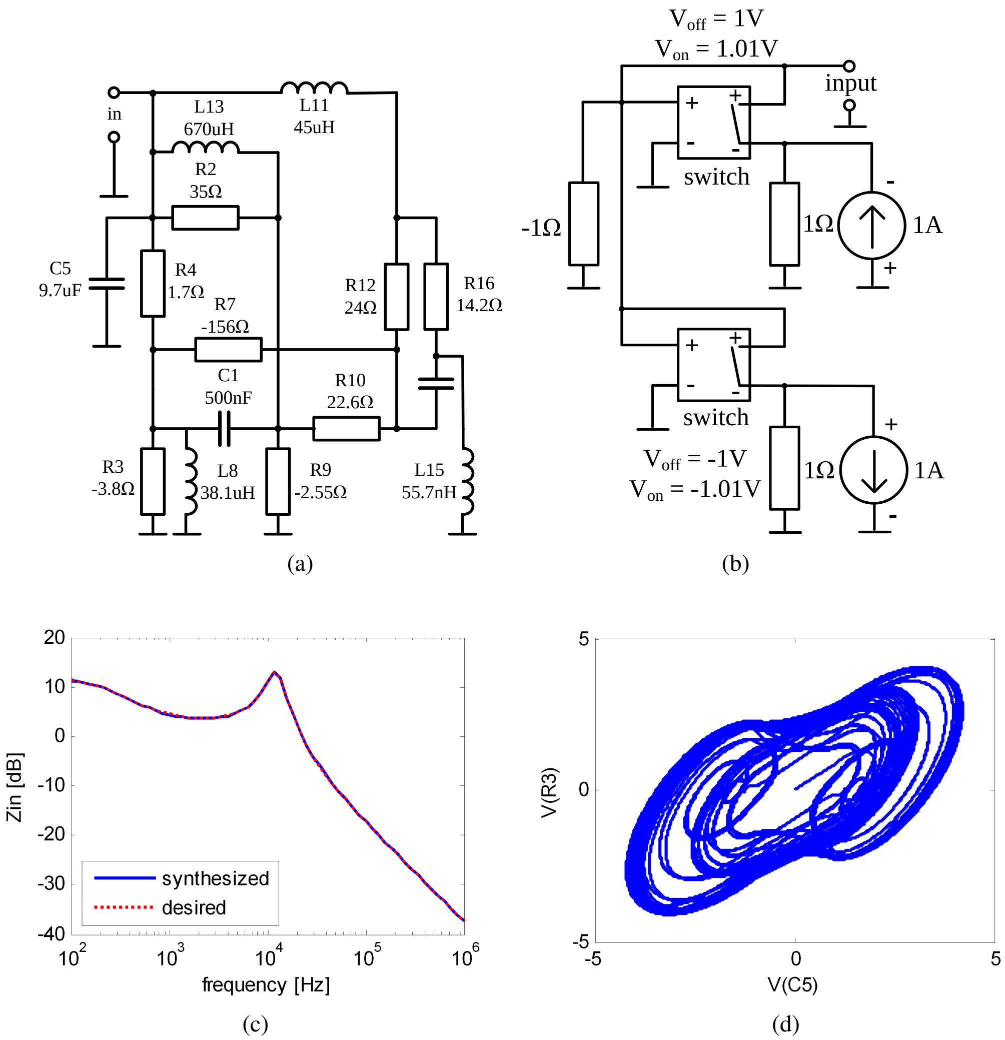
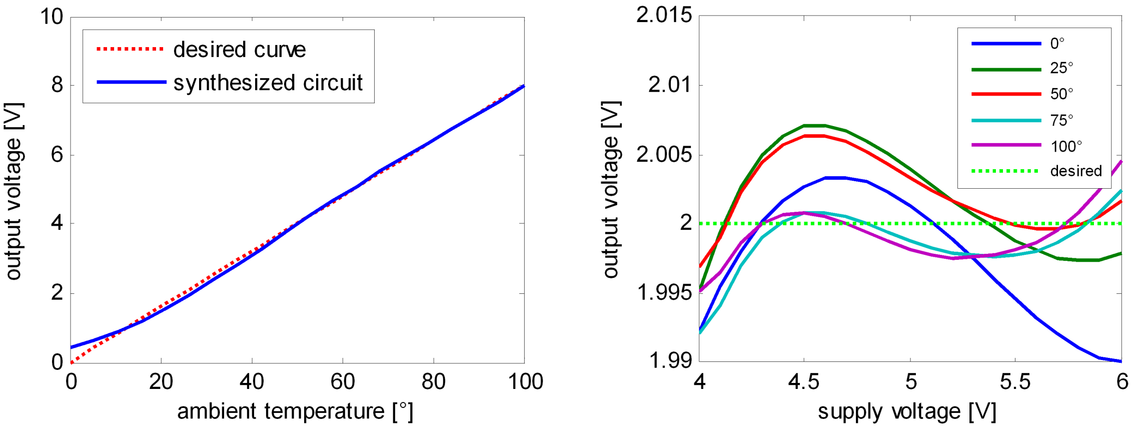


Figure 4: a) Schematic of synthesized admittance network b) PWL circuit c) Z$_{in}$ (3) and Z$_{in}$ of synthesized network d) plane projection V(R3) vs. V(C5).

3.3 SYNTHESIS OF TEMPERATURE SENSOR AND VOLTAGE REFERENCE CIRCUIT

Last two presented problems are synthesis of temperature sensor and synthesis of voltage reference circuit [8]. Output voltage of the temperature sensor is $U_2 = 0.08.T$, where T is ambient temperature of used bipolar transistors. Voltage reference circuit generates constant output voltage 2V for supply voltage 4-6 V and ambient temperature 0 - 100 °C. Both circuits were synthesized using resistors and transistors 2N3904 and 2N3906. Parameters of optimization for temperature sensor: $T_0 = 500, r = 1,6.10^{-5}, T_f = 5.10^{-4}, n = 120, m = 20, evs = 0,93.10^6$. Parameters of optimization for voltage reference: $T_0 = 100, r = 1,6.10^{-5}, T_f = 3.10^{-8}, n = 120, m = 20, evs = 1,4.10^6$. Weights were not used. Cost values were obtained using HSpice. Run times of the optimizations for temperature sensor and voltage reference were 4 days and 19 hours and 5 days and 13 hours respectively. Due to the limited space of this paper circuit schematics are not presented here (temperature sensor consists of 3 resistors, 5 transistors 2N3904 and 7 transistors 2N3906, voltage reference consists of 4 resistors, 9 transistors 2N3904 and 5 transistors 2N3906). In [8] the temperature sensor and the voltage reference were synthesized using AGE method [8]. For obtaining comparable results there was used $5.6.10^6$ and $6.5.10^6$ function evaluations respectively. Output characteristics of both circuits are presented in Figure 5.



(a)

(b)

Figure 5: Output characteristics of a) temperature sensor and b) voltage reference.

4 CONCLUSION

Evolutionary synthesis of analog electronic circuits using simulated annealing method (SA) was described and its capability of synthesis was presented in few selected problems. Proposed method is simple for implementation and has strong ability of search of global optimum. On the other hand there is drawback of long run time what is caused by sequential nature of SA and difficulty of its parallelization. Nevertheless SA has shown good ability of synthesis of analog circuits and for problems of low complexity can be used as good alternative to other methods of evolutionary synthesis of analog electronic circuits.

REFERENCE

[1] Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MIT Press 1992, ISBN 0-262-11170-5

[2] Grimbleby, J. B.: Automatic analogue circuit synthesis using genetic algorithms, IEE Proceedings vol.147, no.6, 2000, pp.319-323, ISSN 1350-2409

[3] Zebulum, R. S., Pacheco, M. A., Vellasco, M. M.: Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms, Florida, CRC Press 2001, ISBN 0-8493-0865-8

[4] Das, A., Vemuri, R.: An automated passive analog circuit synthesis framework using genetic algorithms. In: IEEE Computer Society Annual Symposium on VLSI, 2007, pp.145-152

[5] Krasnicki, M., Phelps, R., Rutenbar, R. A., Carley, L. R.: Maelstrom: Efficient Simulation-Based Synthesis for Custom Analog Cells. In: Design Automation Conference, 1999, pp.945-950

[6] Kirkpatrick, S., Gellat, C., Vecchi, M.: Optimization by simulated annealing, Science, No. 220, 1983, pp. 671-680

[7] Carlson, G. E., Halijak, C. A.: Approximation of Fractional Capacitors by a Regular Newton Process, IEEE Trans. On Circuit Theory, Vol. 11, No. 2, 1964, pp. 210- 213, ISSN 0018-9324.

[8] Mattiussi, C., Floreano, D.: Analog Genetic Encoding for the Evolution of Circuits and Networks, IEEE Transactions on Evolutionary Computation, Vol. 11, No. 5, 2007, pp. 596-607.